

从纵横五个方面精细优化你的Oracle PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/474/2021_2022__E4_BB_8E_E7_BA_B5_E6_A8_AA_E4_c67_474179.htm 在过去的十年中

，Oracle已经成为世界上最专业的数据库之一。对于IT专家来说，就是要确保利用Oracle的强大特性来提高他们公司的生产力。最有效的方法之一是通过Oracle调优。它有大量的调整参数和技术来改进你的Oracle数据库的性能。Oracle调优是一个复杂的主题。关于调优可以写整整一本书，不过，为了改善Oracle数据库的性能，有一些基本的概念是每个Oracle DBA都应该遵从的。在这篇简介中，我们将简要地介绍以下的Oracle主题：

外部调整：我们应该记住Oracle并不是单独运行的。因此我们将查看一下通过调整Oracle服务器以得到高的性能。

Rowre-sequencing以减少磁盘I/O：我们应该懂得Oracle调优最重要的目标是减少I/O.

Oracle SQL调整：Oracle SQL调整是Oracle调整中最重要的领域之一，只要通过一些简单的SQL调优规则就可以大幅度地提升SQL语句的性能，这是一点都不奇怪的。

调整Oracle排序：排序对于Oracle性能也是有很大影响的。

调整Oracle的竞争：表和索引的参数设置对于UPDATE和INSERT的性能有很大的影响。我们首先从调整Oracle外部的环境开始。如果内存和CPU的资源不足的话，任何的Oracle调整都是没有帮助的。

外部的性能问题 Oracle并不是单独运行的。Oracle数据库的性能和外部的环境有很大的关系。这些外部的条件包括有：

- CPUCPU资源的不足令查询变慢。当查询超过了Oracle服务器的CPU性能时，你的数据库性能就受到CPU的限制。
- 内存可用于Oralce的内存数量

也会影响SQL的性能，特别是在数据缓冲和内存排序方面。

网络大量的Net8通信令SQL的性能变慢。许多新手都错误的认为应该首先调整Oracle数据库，而不是先确认外部资源是否足够。实际上，如果外部环境出现瓶颈，再多的Oracle调整都是没有帮助的。在检查Oracle的外部环境时，有两个方面是需要注意的：1、当运行队列的数目超过服务器的CPU数量时，服务器的性能就会受到CPU的限制。补救的方法是为服务器增加额外的CPU或者关闭需要很多处理资源的组件，例如Oracle Parallel Query。2、内存分页。当内存分页时，内存容量已经不足，而内存页是与磁盘上的交换区进行交互的。补救的方法是增加更多的内存，减少Oracle SGA的大小，或者关闭Oracle的多线程服务器。可以使用各种标准的服务器工具来得到服务器的统计数据，例如vmstat，glance，top和sar。DBA的目标是确保数据库服务器拥有足够的CPU和内存资源来处理Oracle的请求。以下让我们来看一下Oracle的row-resequencing是如何能够极大地减少磁盘I/O的。

Row-resequencing（行的重新排序）就象我们上面提到的，有经验的Oracle DBA都知道I/O是响应时间的最大组成部分。其中磁盘I/O特别厉害，因为当Oracle由磁盘上的一个数据文件得到一个数据块时，读的进程就必须等待物理I/O操作完成。磁盘操作要比数据缓冲慢10,000倍。因此，如果可以令I/O最小化，或者减少由于磁盘上的文件竞争而带来的瓶颈，就可以大大地改善Oracle数据库的性能。如果系统响应很慢，通过减少磁盘I/O就可以有一个很快的改善。如果在一个事务中通过按一定的范围搜索primary-key索引来访问表，那么重新以CTAS的方法组织表将是你减少I/O的首要策略。通过在

物理上将行排序为和primary-key索引一样的顺序，就可以加快获得数据的速度。就象磁盘的负载平衡一样，行的重新排序也是很简单的，而且也很快。通过与其它的DBA管理技巧一起使用，就可以在高I/O的系统中大大地减少响应的时间。在高容量的在线事务处理环境中（online transaction processing, OLTP），数据是由一个primary索引得到的，重新排序表格的行就可以令连续块的顺序和它们的primary索引一样，这样就可以在索引驱动表格查询中，减少物理I/O并且改善响应时间。这个技巧仅在应用选择多行的时候有用，或者在使用索引范围搜索和应用发出多个查询来得到连续的key时有效。对于随机的唯一primary-key（主键）的访问将不会由行重新排序中得到好处。让我们看一下它是如何工作的。考虑以下的一个SQL的查询，它使用一个索引来得到100行：

```
0selectsalaryfromemployeewhere last_name like B% ;
```

这个查询将会使用last_name_index，搜索其中的每一行来得到目标行。这个查询将会至少使用100次物理磁盘的读取，因为employee的行存放在不同的数据块中。不过，如果表中的行已经重新排序为和last_name_index的一样，同样的查询又会怎样处理呢？我们可以看到这个查询只需要三次的磁盘I/O就读完全部100个员工的资料（一次用作索引的读取，两次用作数据块的读取），减少了97次的块读取。重新排序带来的性能改善的程度在于在你开始的时候行的乱序性如何，以及你需要由序列中访问多少行。至于一个表中的行与索引的排序键的匹配程度，可以查看数据字典中的dba_indexes和dba_tables视图得到。在dba_indexes的视图中，查看clustering_factor列。如果clustering_factor的值和表中的块数目大致一样，那么你的表

和索引的顺序是一样的。不过，如果clustering_factor的值接近表中的行数，那就表明表格中的行和索引的顺序是不一样的。行重新排序的作用是不可以小看的。在需要进行大范围的索引搜索的大表中，行重新排序可以令查询的性能提高三倍。一旦你已经决定重新排序表中的行，你可以使用以下的工具之一来重新组织表格。使用Oracle的Create Table As Select (CTAS) 语法来拷贝表格 Oracle9i自带的表格重新组织工具 以下，我们来看以下SQL语句的调优。SQL调优

Oracle的SQL调优是一个复杂的主题，甚至是需要整本书来介绍Oracle SQL调优的细微差别。不过有一些基本的规则是每个Oracle DBA都需要跟从的，这些规则可以改善他们系统的性能。SQL调优的目标是简单的：消除不必要的大表全表搜索：不必要的全表搜索导致大量不必要的I/O，从而拖慢整个数据库的性能。调优专家首先会根据查询返回的行数来评价SQL.在一个有序的表中，如果查询返回少于40%的行，或者在一个无序的表中，返回少于7%的行，那么这个查询都可以调整为使用一个索引来代替全表搜索。对于不必要的全表搜索来说，最常见的调优方法是增加索引。可以在表中加入标准的B树索引，也可以加入bitmap和基于函数的索引。要决定是否消除一个全表搜索，你可以仔细检查索引搜索的I/O开销和全表搜索的开销，它们的开销和数据块的读取和可能的并行执行有关，并将两者作对比。在一些情况下，一些不必要的全表搜索的消除可以通过强制使用一个index来达到，只需要在SQL语句中加入一个索引的提示就可以了。在全表搜索是一个最快的访问方法时，将小表的全表搜索放到缓存中，调优专家应该确保有一个专门的数据缓冲用作行缓冲

。在Oracle7中，你可以使用alter table xxx cache语句，在Oracle8或以上，小表可以被强制为放到KEEP池中缓冲。

确保最优的索引使用：对于改善查询的速度，这是特别重要的。有时Oracle可以选择多个索引来进行查询，调优专家必须检查每个索引并且确保Oracle使用正确的索引。它还包

括bitmap和基于函数的索引的使用。 确保最优的JOIN操作：

有些查询使用NESTED LOOP join快一些，有些则是HASH join快一些，另外一些则是sort-merge join更快。这些规则看来简单，不过它们占SQL调优任务的90%，并且它们也无需完全懂得Oracle SQL的内部运作。以下我们来简单概览以下Oracle SQL的优化。我们首先简要查看Oracle的排序，并且看一看排序操作是如何影响性能的。调整Oracle的排序操作

排序是SQL语法中一个小的方面，但很重要，在Oracle的调整中，它常常被忽略。当使用create index、ORDER BY或者GROUP BY的语句时，Oracle数据库将会自动执行排序的操作。通常，在以下的情况下Oracle会进行排序的操作：使用Order by的SQL语句 使用Group by的SQL语句在创建索引的时候进行table join时，由于现有索引的不足而导致SQL优化器调用MERGE SORT. 当与Oracle建立起一个session时，在内存中就会为该session分配一个私有的排序区域。如果该连接是一个专用的连接（dedicated connection），那么就会根据init.ora中sort_area_size参数的大小在内存中分配一个Program Global Area（PGA）。如果连接是通过多线程服务器建立的，那么排序的空间就在large_pool中分配。不幸的是，对于所有的session，用做排序的内存量都必须是一样的，我们不能为需要更大排序的操作分配额外的排序区域。因此

，设计者必须作出一个平衡，在分配足够的排序区域以避免发生大的排序任务时出现磁盘排序（disk sorts）的同时，对于那些并不需要进行很大排序的任务，就会出现一些浪费。当然，当排序的空间需求超出了sort_area_size的大小时，这时将会在TEMP表空间中分页进行磁盘排序。磁盘排序要比内存排序大概慢14,000倍。上面我们已经提到，私有排序区域的大小是有init.ora中的sort_area_size参数决定的。每个排序所占用的大小由init.ora中的sort_area_retained_size参数决定。当排序不能在分配的空间中完成时，就会使用磁盘排序的方式，即在Oracle实例中的临时表空间中进行。磁盘排序的开销是很大的，有几个方面的原因。首先，和内存排序相比较，它们特别慢；而且磁盘排序会消耗临时表空间中的资源。Oracle还必须分配缓冲池块来保持临时表空间中的块。无论什么时候，内存排序都比磁盘排序好，磁盘排序将会令任务变慢，并且会影响Oracle实例的当前任务的执行。还有，过多的磁盘排序将会令free buffer waits的值变高，从而令其它任务的数据块由缓冲中移走。接着，让我们看一下Oracle的竞争，并且看一下表的存储参数的设置是如何影响SQL UPDATE和INSERT语句的性能的。调整Oracle的竞争 Oracle的其中一个优点时它可以管理每个表空间中的自由空间。Oracle负责处理表和索引的空间管理，这样就可以让我们无需懂得Oracle的表和索引的内部运作。不过，对于有经验的Oracle调优专家来说，他需要懂得Oracle是如何管理表的extent和空闲的数据块。对于调整拥有高的insert或者0update的系统来说，这是非常重要的。要精通对象的调整，你需要懂得freelists和freelist组的行为，它们和pctfree及pctused参数的值有关。这些知识对于企业资源计

划（ERP）的应用是特别重要的，因为在这些应用中，不正确的表设置通常是DML语句执行慢的原因。对于初学者来说，最常见的错误是认为默认的Oracle参数对于所有的对象都是最佳的。除非磁盘的消耗不是一个问题，否则在设置表的pctfree和pctused参数时，就必须考虑平均的行长和数据库的块大小，这样空的块才会被有效地放到freelists中。当这些设置不正确时，那些得到的freelists也是"dead"块，因为它们没有足够的空间来存储一行，这样将会导致明显的处理延迟。Freelists对于有效地重新使用Oracle表空间中的空间是很重要的，它和pctfree及pctused这两个存储参数的设置直接相关。通过将pctused设置为一个高的值，这时数据库就会尽快地重新使用块。不过，高性能和有效地重新使用表的块是对立的。在调整Oracle的表格和索引时，需要认真考虑究竟需要高性能还是有效的空间重用，并且据此来设置表的参数。以下我们来看一下这些freelists是如何影响Oracle的性能的。当有一个请求需要插入一行到表格中时，Oracle就会到freelist中寻找一个有足够的空间来容纳一行的块。你也许知道，freelist串是放在表格或者索引的第一个块中，这个块也被称为段头（segment header）。pctfree和pctused参数的唯一目的就是为控制块如何在freelists中进出。虽然freelist link和unlink是简单的Oracle功能，不过设置freelist link（pctused）和unlink（pctfree）对Oracle的性能确实有影响。由DBA的基本知识知道，pctfree参数是控制freelist un-links的（即将块由freelists中移除）。设置pctfree=10意味着每个块都保留10%的空间用作行扩展。pctused参数是控制freelist re-links的。设置pctused=40意味着只有在块的使用低于40%时才会回到表格的freelists中。

许多新手对于一个块重新回到freelists后的处理都有些误解。其实，一旦由于一个删除的操作而令块被重新加入到freelist中，它将会一直保留在freelist中即使空间的使用超过了60%，只有在到达pctfree时才会将数据块由freelist中移走。表格和索引存储参数设置的要求总结以下的一些规则是用来设置freelists，freelistgroups，pctfree和pctused存储参数的。你也知道，pctused和pctfree的值是可以很容易地通过alter table命令修改的，一个好的DBA应该知道如何设置这些参数的最佳值。有效地使用空间和高性能之间是有矛盾的，而表格的存储参数就是控制这个方面的矛盾：对于需要有效地重新使用空间，可以设置一个高的pctused值，不过副作用是需要额外的I/O.一个高的pctused值意味着相对满的块都会放到freelist中。因此，这些块在再次满之前只可以接受几行记录，从而导致更多的I/O. 追求高性能的话，可以将pctused设置为一个低的值，这意味着Oracle不会将数据块放到freelists中直到它几乎是空的。那么块将可以在满之前接收更多的行，因此可以减少插入操作的I/O.要记住Oracle扩展新块的性能要比重新使用现有的块高。对于Oracle来说，扩展一个表比管理freelists消耗更少的资源。让我们来回顾一下设置对象存储参数的一些常见规则：经常将pctused设置为可以接收一条新行。对于不能接受一行的free blocks对于我们来说是没有用的。如果这样做，将会令Oracle的性能变慢，因为Oracle将在扩展表来得到一个空的块之前，企图读取5个"dead"的free block. 表格中chained rows的出现意味着pctfree太低或者是db_block_size太少。在很多情况下，RAW和LONG RAW列都很巨大，以至超过了Oracle的最大块的大小，这时chained rows是不可以避免

的。 如果一个表有同时插入的SQL语句，那么它需要有同时删除的语句。运行单一个一个清除的工作将会把全部的空闲块放到一个freelist中，而没有其它包含有任何空闲块的freelists出现。 freelist参数应该设置为表格同时更新的最大值。例如，如果在任何时候，某个表最多有20个用户执行插入的操作，那么该表的参数应该设置为freelists=20. 应记住的是freelist groups参数的值只是对于Oracle Parallel Server和Real Application Clusters才是有用的。对于这类Oracle，freelist groups应该设置为访问该表格的Oracle Parallel Server实例的数目。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com